



00862.023114.

PATENT APPLICATION

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:)	
	:	Examiner: N.Y.A.
TADAYOSHI NAKAYAMA)	
	:	Group Art Unit: N.Y.A.
Application No.: 10/603,603)	
	:	
Filed: June 26, 2003)	
	:	
For: HADAMARD TRANSFORMATION)	
METHOD AND DEVICE	:	September 16, 2003

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

SUBMISSION OF PRIORITY DOCUMENT

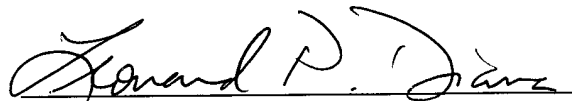
Sir:

In support of Applicant's claim for priority under 35 U.S.C. § 119, enclosed
is a certified copy of the following foreign application:

JAPAN 2002-193298, filed July 2, 2002.

Applicant's undersigned attorney may be reached in our New York office by telephone at (212) 218-2100. All correspondence should continue to be directed to our address given below.

Respectfully submitted,


Attorney for Applicant

Registration No. 28,286

FITZPATRICK, CELLA, HARPER & SCINTO
30 Rockefeller Plaza
New York, New York 10112-3801
Facsimile: (212) 218-2200

NY_MAIN 370833v1

日 本 国 特 許 庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日 2 0 0 2 年 7 月 2 日
Date of Application:

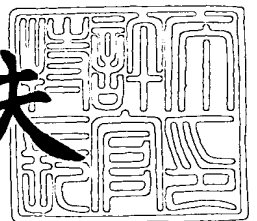
出 願 番 号 特 願 2 0 0 2 - 1 9 3 2 9 8
Application Number:
[ST. 10/C] : [J P 2 0 0 2 - 1 9 3 2 9 8]

出 願 人 キヤノン株式会社
Applicant(s):

2 0 0 3 年 7 月 2 2 日

特許庁長官
Commissioner,
Japan Patent Office

今 井 康 夫



【書類名】 特許願

【整理番号】 4732004

【提出日】 平成14年 7月 2日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 5/00

【発明の名称】 アダマール変換処理方法及びその装置

【請求項の数】 11

【発明者】

 【住所又は居所】 東京都大田区下丸子 3 丁目 3 0 番 2 号 キヤノン株式会社
社内

 【氏名】 中山 忠義

【特許出願人】

 【識別番号】 000001007

 【氏名又は名称】 キヤノン株式会社

【代理人】

 【識別番号】 100076428

 【弁理士】

 【氏名又は名称】 大塚 康德

 【電話番号】 03-5276-3241

【選任した代理人】

 【識別番号】 100112508

 【弁理士】

 【氏名又は名称】 高柳 司郎

 【電話番号】 03-5276-3241

【選任した代理人】

 【識別番号】 100115071

 【弁理士】

 【氏名又は名称】 大塚 康弘

 【電話番号】 03-5276-3241

【選任した代理人】**【識別番号】** 100116894**【弁理士】****【氏名又は名称】** 木村 秀二**【電話番号】** 03-5276-3241**【手数料の表示】****【予納台帳番号】** 003458**【納付金額】** 21,000円**【提出物件の目録】****【物件名】** 明細書 1**【物件名】** 図面 1**【物件名】** 要約書 1**【包括委任状番号】** 0102485**【プルーフの要否】** 要

【書類名】 明細書

【発明の名称】 アダマール変換処理方法及びその装置

【特許請求の範囲】

【請求項 1】 各アダマール変換処理ユニットのそれぞれにおいて、入力信号に対してアダマール変換行列による変換処理を行う第 1 変換工程と、

前記第 1 変換工程により各アダマール変換処理ユニットから出力される変換データの内、奇数個のデータの各最下位ビットを切り上げ処理し、残りの奇数個のデータの各最下位ビットを切り捨て処理して整数化する第 1 丸め処理工程と、

前記アダマール変換処理ユニットの数と同数のアダマール変換処理ユニットを用いて、前記第 1 変換工程の後段において、前記第 1 丸め処理工程で、入力信号のうち奇数個が、それと同数のアダマール変換処理ユニットの出力を前記切り上げ処理したもの、残りの奇数個は他の残りのアダマール変換処理ユニットの出力を前記切り捨て処理した信号を各アダマール変換処理ユニットに入力して、各アダマール変換処理ユニットのそれぞれにおいて、入力信号に対してアダマール変換行列による変換処理を行う第 2 変換工程と、

前記第 2 変換工程により各アダマール変換処理ユニットから出力される変換データの内、奇数個のデータの各最下位ビットを切り上げ処理し、残りの奇数個のデータの各最下位ビットを切り捨て処理して整数化する第 2 丸め処理工程と、
を有することを特徴とするアダマール変換処理方法。

【請求項 2】 4 つの 4 点アダマール変換処理ユニットのそれぞれにおいて、入力信号に対してアダマール変換行列による変換処理を行う第 1 変換工程と、

前記第 1 変換工程により各 4 点アダマール変換処理ユニットから出力される 4 つの変換データの内、奇数個のデータの各最下位ビットを切り上げ処理し、残りの奇数個のデータの各最下位ビットを切り捨て処理して整数化する第 1 丸め処理工程と、

4 つの 4 点アダマール変換処理ユニットを用いて、前記第 1 変換工程の後段において、前記第 1 丸め処理工程で、入力信号のうち奇数個が 4 点アダマール変換処理ユニットの出力を前記切り上げ処理したもの、残りの奇数個は他のロスレス 4 点アダマール変換処理ユニットの出力を前記切り捨て処理した信号を各 4 点ア

ダマール変換処理ユニットに入力して、4つの4点アダマール変換処理ユニットのそれぞれにおいて、入力信号に対して4点アダマール変換行列による変換処理を行う第2変換工程と、

前記第2変換工程により各4点アダマール変換処理ユニットから出力される変換データの内、奇数個のデータの各最下位ビットを切り上げ処理し、残りの奇数個のデータの各最下位ビットを切り捨て処理して整数化する第2丸め処理工程と、

を有することを特徴とするアダマール変換処理方法。

【請求項3】 複数のアダマール変換処理ユニットを具備し、各処理ユニットにおいて、入力信号に対してアダマール変換行列による変換処理を行う第1変換手段と、

前記第1変換手段により各アダマール変換処理ユニットから出力される変換データの内、奇数個のデータの各最下位ビットを切り上げ処理し、残りの奇数個のデータの各最下位ビットを切り捨て処理して整数化する第1丸め処理手段と、

前記複数のアダマール変換処理ユニットの数と同数のアダマール変換処理ユニットを具備し、前記第1変換手段の後段において、前記第1丸め処理手段で、入力信号のうち奇数個が、それと同数のアダマール変換処理ユニットの出力を前記切り上げ処理したもの、残りの奇数個は他の残りのアダマール変換処理ユニットの出力を前記切り捨て処理した信号を各アダマール変換処理ユニットに入力して、各アダマール変換処理ユニットのそれぞれにおいて、入力信号に対してアダマール変換行列による変換処理を行う第2変換手段と、

前記第2変換手段により各アダマール変換処理ユニットから出力される変換データの内、奇数個のデータの各最下位ビットを切り上げ処理し、残りの奇数個のデータの各最下位ビットを切り捨て処理して整数化する第2丸め処理手段と、
を有することを特徴とするアダマール変換処理装置。

【請求項4】 4つの4点アダマール変換処理ユニットを具備し、各処理ユニットにおいて、入力信号に対してアダマール変換行列による変換処理を行う第1変換手段と、

前記第1変換手段により各4点アダマール変換処理ユニットから出力される4

つの変換データの内、奇数個のデータの各最下位ビットを切り上げ処理し、残りの奇数個のデータの各最下位ビットを切り捨て処理して整数化する第1丸め処理手段と、

4つの4点アダマール変換処理ユニットを用いて、前記第1変換手段の後段において、前記第1丸め処理手段で、入力信号のうち奇数個が4点アダマール変換処理ユニットの出力を前記切り上げ処理したもの、残りの奇数個は他の4点アダマール変換処理ユニットの出力を前記切り捨て処理した信号を各4点アダマール変換処理ユニットに入力して、4つの4点アダマール変換処理ユニットのそれぞれにおいて、入力信号に対して4点アダマール変換行列による変換処理を行う第2変換手段と、

前記第2変換手段により各4点アダマール変換処理ユニットから出力される変換データの内、奇数個のデータの各最下位ビットを切り上げ処理し、残りの奇数個のデータの各最下位ビットを切り捨て処理して整数化する第2丸め処理手段と、

を有することを特徴とするアダマール変換処理装置。

【請求項5】 4×4のブロックデータを水平方向と垂直方向に2次的に処理するアダマール変換処理装置であって、

入力データを4個単位でアダマール変換処理する第1変換部と、

ブロックデータの列番号或いは行番号に基づいて丸め処理用の加算データを生成する第1デコーダと、

前記第1変換部の出力に前記第1デコーダからの前記加算データを加算する第1加算部と、

前記第1加算部の出力の最下位の1ビットを切り捨てる第1切り捨て部と、

前記第1変換部と前記第1デコーダと前記第1加算部と前記第1切り捨て部とで処理したデータを転置するためのバッファと、

前記バッファの出力を入力し4個単位でアダマール変換処理する第2変換部と

ブロックデータの列番号或いは行番号に基づいて丸め処理用の加算データを生成する第2デコーダと、

前記第 2 変換部の出力に前記第 2 デコーダからの前記加算データ出力を加算する第 2 加算部と、

前記第 2 加算部の出力の最下位の 1 ビットを切り捨てる第 2 切り捨て部と、
を有することを特徴とするアダマール変換処理装置。

【請求項 6】 前記第 1 及び第 2 デコーダは、0 または 0.5 の 4 つのデータを出力することを特徴とする請求項 5 に記載のアダマール変換処理装置。

【請求項 7】 前記第 1 及び第 2 デコーダは、対応する変換部が出力するデータの最下位ビットを受け取り、前記最下位ビットが 1 の時は 0.5 または -0.5 の 4 つのデータを、前記最下位ビットが 0 の時は 4 つとも 0 を出力することを特徴とする請求項 5 に記載のアダマール変換処理装置。

【請求項 8】 前段のアダマール変換処理と後段のその他の変換処理とを有する 1 次元離散コサイン変換処理を 2 段階適用して 2 次元処理する際に、各部の処理順序を変えて、前記前段のアダマール変換をまとめて 2 次元処理した後に、前記後段の変換処理をまとめて 2 段階処理することを特徴とする 2 次元 DCT 変換処理方法。

【請求項 9】 前段のロスレスアダマール変換部と後段のその他のロスレス変換部とを具備する 1 次元離散コサイン変換を 2 段階用いて 2 次元処理する際に、各部の処理順序を変えて、前記前段のアダマール変換部をまとめて 2 次元処理した後に、前記後段のロスレス変換処理をまとめて 2 段階処理するようにしたことを特徴とする 2 次元 DCT 変換処理方法。

【請求項 10】 請求項 1, 2, 8, 9 のいずれか 1 項に記載のアダマール変換処理方法を実行することを特徴とするプログラム。

【請求項 11】 請求項 10 に記載のアダマール変換処理方法を実行するプログラムを記憶したことを特徴とする、コンピュータにより読み取り可能な記憶媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、整数データを出力する可逆変換が可能なアダマール変換処理方法及

びその装置に関するものである。

【0002】

【従来の技術】

画像、特に多値画像は非常に多くの情報を含んでおり、その画像を蓄積或いは伝送する際には、そのデータ量が問題となる。このため画像の蓄積或いは伝送に際しては、その多値画像の持つ冗長性を除いたり、或いは画質の劣化が視覚的に認識し難い程度で、その多値画像の内容を変更することによって符号化して、そのデータ量を削減する高能率符号化が用いられる。例えば、静止画像の国際標準符号化方式としてISOとITU-Tにより勧告されたJPEGでは、画像データをブロックごと（8画素×8画素）に離散コサイン変換（DCT）してDCT係数に変換した後に、各DCT係数をそれぞれ量子化し、更に、その量子化した値をエントロピー符号化することにより画像データを圧縮している。このDCTを利用した圧縮技術には、このJPEG以外に、H261, MPEG1/2/4等がある。

【0003】

このDCT変換に関連する処理として、或いは画像データ変換する処理としてアダマール変換がある。このアダマール変換は、変換行列の要素が「1」又は「-1」のみからなる直交変換で、加算と減算のみで実現できる最もシンプルな直交変換である。

【0004】

このアダマール変換のうち、2点アダマール変換の変換行列H2は、以下のよう

【0005】

【数1】

$$H_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad (1)$$

【0006】

一般的なN（=2ⁿ）点アダマール変換行列HNは、（N/2）点アダマール変換行列HN/2と上記2点アダマール変換行列H2との間のクロネッカー積で再帰的

に定義することができる。

【 0 0 0 7 】

【数 2】

$$\begin{aligned} H_N &= H_{N/2} \otimes H_2 \\ &= \frac{1}{\sqrt{2}} \begin{bmatrix} H_{N/2} & H_{N/2} \\ H_{N/2} & -H_{N/2} \end{bmatrix} \end{aligned} \quad (2)$$

【 0 0 0 8 】

上記定義から 4 点アダマール変換行列を求めると、

【 0 0 0 9 】

【数 3】

$$H_4 = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \quad (3)$$

となる。この 4 点アダマール変換行列はナチュラル型と呼ばれるもので、基底ベクトルがシーケンシの順番に並んでいない。そこで、基底ベクトルの置換を繰り返して 2 行目の基底ベクトルを 4 行目に移動すると、基底ベクトルの順序がシーケンシ順序の変換行列 WH_4 となる。

【 0 0 1 0 】

【数 4】

$$WH_4 = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \quad (4)$$

【 0 0 1 1 】

この変換行列 WH_4 は、ウォルシュ型或いはウォルシュアダマール変換行列と呼ばれている。アダマール変換は可逆な直交変換であることが知られており、上記ナチュラル型、ウォルシュ型のいずれも可逆な変換が可能である上、変換行列が対称行列になっている。

【0012】

前述したように、アダマール変換は可逆変換であると一般的に言われているが、これは数学的に可逆であるということである。即ち、変換時及び逆変換において演算誤差が発生しないことを前提としており、そのためのデータ形式として固定小数点或いは浮動小数点で演算を行なう必要がある。そして変換処理後の有効桁数を全て保持する必要がある。

【0013】

しかし、圧縮符号化を前提としたアダマール変換では、変換処理後に有効桁数を少しでも減らしたいという要望がある。具体的には、整数入力の変換処理して発生する小数点以下のデータは、明らかに入力データより増加した桁数（情報）と捉えることができるので、この小数点以下のデータを無くすことはデータ圧縮のためには不可欠である。ところが、この小数点以下のデータを単純に丸め処理してしまうと可逆性は損なわれてしまう。例えば、

123, 78, 84, 56

の4つのデータを（4）式の変換行列でアダマール変換すると以下のようにになる。

【0014】

$170.5 (= (123+78+84+56)/2)$, $30.5 (= (123+78-84-56)/2)$, $8.5 (= (123-78-84+56)/29)$, $36.5 (= (123-78+84-56)/2)$

これらのそれぞれを単純に四捨五入して整数化すると、

171 31, 9, 37

となり、これを逆変換（逆変換で用いる行列は変換行列の転置行列であり、これは変換行列と同じである）すると、

124, 78, 84, 56

となる。このように先頭データの「123」が、上述のアダマール変換及びその逆変換により「124」になってしまっている。このように、整数化したデータを入力するアダマール変換では可逆性を保証できない。以下では、整数化したデータを入力するアダマール変換を整数型アダマール変換と呼び、可逆変換が可能な整数型アダマール変換を整数型可逆アダマール変換或いはロスレスアダマール

変換と呼ぶことにする。

【0015】

従来、ロスレス4点アダマール変換は、以下のように実現していた。

①4点アダマール変換行列H4を、対角要素が「1」となる三角行列の積に分解する。

②元の変換行列に行の入れ替え操作Qを加える。

③上記①②から、QH4を以下の行列積の形に分解する。

【0016】

【数5】

$$QH_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ -1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1/2 & 1/2 \\ 0 & 1 & 1/2 & -1/2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix} \quad (5)$$

④この変換をシグナルフローで表現する（図1のように表わされる）。

⑤このシグナルフロー中の乗算処理で発生する小数点以下のデータを丸めて整数化する。

【0017】

以上の①～⑤により実現する。以下に若干の補足説明を加える。

【0018】

図1のシグナルフローにおいて、可逆変換処理を実現する上でよく用いられる梯子型ネットワーク（Ladder Network）構成になっている。このLadder Networkにおいて、小数点以下のデータが発生する乗算器100、101の出力側に整数化を図るための丸め処理（図2の200、201）を導入することで、整数型データを出力する変換処理の可逆化が（可逆変換処理の分野では）一般的な手法として用いられている。

【0019】

尚、ここで、変換処理と逆変換処理の間で対応する乗算出力の丸め処理が同一であれば、丸め処理200、201の内容はなんでもよい。

【0020】

図1のシグナルフローに丸め処理を導入して可逆化した（可逆変換を可能にした）シグナルフローを図2に示す。これが、整数型可逆4点アダマール変換を実現する従来の演算方法であった。

【0021】

【発明が解決しようとする課題】

このようなロスレス整数型可逆4点アダマール変換を用いて、それを多段接続することによって、ロスレス16点アダマール変換を構成できる。しかし、各変換出力の誤差（数学的な変換とロスレス変換との差）の平均値や平均二乗誤差にばらつきが発生し、精度の良い変換係数と精度の悪い変換係数とができてしまう。

【0022】

また、従来のロスレス2次元DCT（離散コサイン変換）は、ロスレスでない従来型の2次元DCTを基準とした時の変換時の誤差が大きかった。

【0023】

本発明は上記従来例に鑑みてなされたもので、ロスレスアダマール変換処理によって発生する平均誤差が理論上ゼロになるように丸め処理の内容を定めることにより変換係数の精度を高めたアダマール変換処理方法及びその装置を提供することを目的とする。

【0024】

【課題を解決するための手段】

上記目的を達成するために本発明のアダマール変換処理方法は以下のような工程を備える。即ち、

各アダマール変換処理ユニットのそれぞれにおいて、入力信号に対してアダマール変換行列による変換処理を行う第1変換工程と、

前記第1変換工程により各アダマール変換処理ユニットから出力される変換データの内、奇数個のデータの各最下位ビットを切り上げ処理し、残りの奇数個のデータの各最下位ビットを切り捨て処理して整数化する第1丸め処理工程と、

前記アダマール変換処理ユニットの数と同数のアダマール変換処理ユニットを用いて、前記第1変換工程の後段において、前記第1丸め処理工程で、入力信号

のうち奇数個が、それと同数のアダマール変換処理ユニットの出力を前記切り上げ処理したもの、残りの奇数個は他の残りのアダマール変換処理ユニットの出力を前記切り捨て処理した信号を各アダマール変換処理ユニットに入力して、各アダマール変換処理ユニットのそれぞれにおいて、入力信号に対してアダマール変換行列による変換処理を行う第2変換工程と、

前記第2変換工程により各アダマール変換処理ユニットから出力される変換データの内、奇数個のデータの各最下位ビットを切り上げ処理し、残りの奇数個のデータの各最下位ビットを切り捨て処理して整数化する第2丸め処理工程と、
を有することを特徴とする。

【0025】

上記目的を達成するために本発明のアダマール変換処理装置は以下のような構成を備える。即ち、

複数のアダマール変換処理ユニットを具備し、各処理ユニットにおいて、入力信号に対してアダマール変換行列による変換処理を行う第1変換手段と、

前記第1変換手段により各アダマール変換処理ユニットから出力される変換データの内、奇数個のデータの各最下位ビットを切り上げ処理し、残りの奇数個のデータの各最下位ビットを切り捨て処理して整数化する第1丸め処理手段と、

前記複数のアダマール変換処理ユニットの数と同数のアダマール変換処理ユニットを具備し、前記第1変換手段の後段において、前記第1丸め処理手段で、入力信号のうち奇数個が、それと同数のアダマール変換処理ユニットの出力を前記切り上げ処理したもの、残りの奇数個は他の残りのアダマール変換処理ユニットの出力を前記切り捨て処理した信号を各アダマール変換処理ユニットに入力して、各アダマール変換処理ユニットのそれぞれにおいて、入力信号に対してアダマール変換行列による変換処理を行う第2変換手段と、

前記第2変換手段により各アダマール変換処理ユニットから出力される変換データの内、奇数個のデータの各最下位ビットを切り上げ処理し、残りの奇数個のデータの各最下位ビットを切り捨て処理して整数化する第2丸め処理手段と、
を有することを特徴とする。

【0026】

【発明の実施の形態】

以下、添付図面を参照して本発明の好適な実施の形態を詳細に説明する。

【0 0 2 7】**<実施の形態 1>**

図 3 は、従来のロスレス 1 6 点アダマール変換の構成を示す図である。

【0 0 2 8】

図 3 に示すように、ロスレス 1 6 点アダマール変換処理部 3 0 0 ～ 3 0 7 を多段に 8 個接続している。ここでは 1 6 個の入力データ（d0～d15）を 4 つずつに分け、それぞれをロスレス 4 点アダマール変換した後に、同じレベルの変換係数を 4 つ集めて、再度ロスレス 4 点アダマール変換を行なうと、ロスレス 1 6 点アダマール変換となる。

【0 0 2 9】

ここで、このロスレス 4 点アダマール変換部 3 0 0 ～ 3 0 7 に、図 2 に示す構成のアダマール変換部を採用することができる。

【0 0 3 0】

この 8 種類のそれぞれの丸め処理を簡単に説明すると、4 つの出力の内、奇数個を切り上げ、残りの奇数個を切り捨て処理するものである。

【0 0 3 1】

図 4 は、これら 8 種類のロスレス 4 点アダマール変換部 3 0 0 ～ 3 0 7 のそれぞれにおける処理を説明する図である。ここでは前段の 4 点アダマール変換部 3 0 0 ～ 3 0 7 のそれぞれで上述の（3）式又は（4）式のアダマール変換行列による変換処理を行ない、後段の 4 0 0 ～ 4 0 7 で丸め処理を行なう。ここでは丸め処理の内容を簡潔に「+」と「-」で示しているが、ここで「+」は切り上げ、「-」は切り捨てを意味する。なお、本発明で言う切り上げ・切り捨てとは、絶対値表現ではなく 2 の補数表現上での切り上げ・切り捨て処理のことを言う。例えば、-7.5 の補数表現は 1 1 1 1 1 0 0 0.1 となり、この小数部を切り上げると 1 1 1 1 1 0 0 1（-7）となり、切り捨てると 1 1 1 1 1 0 0 0（-8）となる。

【0 0 3 2】

逆変換も一種のロスレス変換である。何故なら、逆変換後のデータを又再変換すると逆変換前のデータになるからである。よって、図 4 に示した 8 種類は逆変換の全種類でもある。但し、アダマール変換行列の種類（ナチュラル型・ウォルシュ型）によって、変換と逆変換の対応が異なる。

【 0 0 3 3 】

ナチュラル型では、変換と逆変換の関係は以下のようになる。

(a) → (a) (b) → (g) (c) → (f) (d) → (h)
 (e) → (e) (f) → (c) (g) → (b) (h) → (d)

ウォルシュ型では、変換と逆変換の関係は下記のようにになる。

(a) → (a) (b) → (f) (c) → (h) (d) → (g)
 (e) → (e) (f) → (b) (g) → (d) (h) → (c)

これ以降、特に断わらない限り、ウォルシュ型の変換行列を使用した場合で説明する。

【 0 0 3 4 】

図 4 の 8 種類の丸め処理のいずれを図 3 に用いても、ロスレス 1 6 点アダマール変換であることに変わりはない。その組み合わせは 8^8 個もの種類が存在する。そこで、次の課題を課す。

(A 1) 各変換係数の平均誤差がゼロになるようにする。

【 0 0 3 5 】

但し、入力データとして相関の無い乱数を仮定する。この課題 (A 1) を満足するロスレス 1 6 点アダマール変換は以下の条件 (B 1)、(B 2) に基づけば構成できる。

(B 1) 2 段目（後段）の各アダマール変換部 3 0 4 ～ 3 0 7 に入力する 4 つの中間データは、奇数個が切り上げ処理し、残りの奇数個が切り捨て処理したものになるように 1 段目の丸め処理を決める。この時の 1 段目の丸め処理は当然前記 8 種類の丸め処理の 1 つとなるようにする。

(B 2) 2 段目の各アダマール変換部 3 0 4 ～ 3 0 7 は、4 つの入力データが同一のアダマール変換から来たものとして捉え、逆変換となるよう丸め処理の種類を決定する。

【0036】

1 段目の 4 つのアダマール変換部 300～303 の内、3 つまでは自由に丸め処理を決められるが、残りの 1 つは条件 (B1) を満たそうとすると一意に決まってしまう。このように、条件 (B1) を満たす構成は簡単に得られ、その種類は、 $512 (= 8^3)$ 種類となる。

【0037】

図 5 は、本発明の実施の形態 1 に係るロスレス 16 点アダマール変換の構成を示すブロック図である。尚、図 5 における 4 点アダマール変換部 501～504、521～524 は図 3 及び図 4 の 4 点アダマール変換部 300～307 と同じ構成である。

【0038】

図 5 において、丸め処理 511～514、531～534 における「+」は切り上げを示し、「-」は切り捨て処理を示している。この例では全 8 種類のうち 7 種類の丸め演算を使用している。このように構成した場合、どの出力の平均誤差も「0」になるだけでなく、誤差の分布まで同じになるため、平均二乗誤差も同じになる。以下では、誤差の分布を通して平均誤差が「0」になる理由について説明する。

【0039】

図 5 における 2 段目のアダマール変換部 521～524 のそれぞれの 4 つの入力 (中間データ) は、同一のアダマール変換部から入力されたものではないため相関が無い。例えば、アダマール変換部 521 の入力データにおける重畳誤差は、 $(0, 0, 0, 0) \sim (+0.5, -0.5, -0.5, -0.5)$ の全 16 通りが各々等しい確立で発生する。図 5 において、中間データ 540 は切り上げ処理されたデータであるが、 $1/2$ の確立で切り上げによる誤差 $(+0.5)$ が発生し、 $1/2$ の確率で、その変換結果が整数の誤差の無いデータとなる。他の 3 つの中間データ 541～543 は、その丸め処理が切り捨てであるだけで、 $1/2$ の確立で切り捨てによる誤差 (-0.5) が発生し、 $1/2$ の確率で、その変換結果が整数の誤差の無いデータとなる点は同じである。

【0040】

この重畳誤差と (4) 式の変換行列の第 1 行との内積に係数 $(1/2)$ を掛けた値は、最小「 -0.75 」から「 0.25 」刻みで最大「 0.25 」まで分布する。更に詳しくみると、「 -0.75 」, 「 -0.5 」, 「 -0.25 」, 「 0 」, 「 $+0.25$ 」という値が、各々確立「 $1/16$ 」, 「 $1/4$ 」, 「 $3/8$ 」, 「 $1/4$ 」, 「 $1/16$ 」で分布する。この分布を示したのが図 6 (a) である。これに、2 段目のアダマール変換部 521 による変換処理後の丸め処理 531 で「 $+0.5$ 」が付加されると、誤差の分布は図 6 (b) のようになる。

【0041】

図 5 の 2 段目のアダマール変換部 521 による変換処理後の丸め処理 531 で「 $+0.5$ 」が付加される確立は、最下位ビットが「1」になる確立であり、これは「 $1/2$ 」である。

【0042】

図 6 (a) と図 6 (b) に示す分布がそれぞれ「 $1/2$ 」の確立で発生する場合、全体の誤差分布は図 6 (c) のようになる。また、この分布の平均二乗誤差は「 0.125 」である。

【0043】

また (4) 式の変換行列の第 2, 第 3, 第 4 行との内積に「 $1/2$ 」を掛けた値は、上記分布とは対称に「 -0.25 」から「 0.75 」の間で「 0.25 」刻みで分布する。2 段目のアダマール変換部の後の丸め処理で付加される値も「 -0.5 」と対称であるため、全体の誤差分布も対称となり、結局、同じ分布になる。他の 3 つのアダマール変換の入力データは、重畳誤差の分布が少し異なるが、同様に計算すると同じ誤差の分布となり、平均誤差は「 0 」、平均二乗誤差は「 0.125 」となる。結局、図 5 の全ての出力 $y_0 \sim y_{15}$ において、平均誤差は「 0 」、平均二乗誤差は「 0.125 」となる。

【0044】

参考に、上記条件 (B1) を満たさない構成を図 7 に示し、その平均誤差と平均二乗誤差がどうなるかを説明する。

【0045】

図7では、2段目のアダマール変換部521の4つの入力データは全て切り上げ処理されたデータが入力されているためその重畳誤差は、 $(0, 0, 0, 0) \sim (+0.5, +0.5, +0.5, +0.5)$ と+方向に偏っている。そのため、上記(4)式の変換行列の第1行との内積に係数 $(1/2)$ を掛けた値は図8(a)のような分布を示す。この後、丸め処理711で更に「+0.5」を付加すると、誤差の分布は図8(b)のようになり、全体の誤差分布は図8(c)のようになる。よって、出力y0の平均誤差は「0.75」、平均二乗誤差は「0.6875」となる。しかし、誤差の分布の仕方は図6(c)と同じなので、その分散は「0.125」である。ここで平均二乗誤差は、 $\{(\text{平均誤差})^2 + \text{分散}\}$ なので、平均誤差が「0」でない分だけ平均二乗誤差が大きくなってしまふ。

【0046】

図7では、出力y0～y15のうちのどの出力に重畳する誤差もその分布の仕方は同じであり、その分散は「0.125」である。このように出力によって平均誤差が異なり、それに応じて平均二乗誤差も変わってくる。全ての出力を平均誤差と平均二乗誤差で分類すると、下記の3つのグループに分けられる。

【0047】

平均誤差 = 0.75、 平均二乗誤差 = 0.6875 : y0
 平均誤差 = 0.25、 平均二乗誤差 = 0.1875 : y7, y8, y15
 平均誤差 = -0.25、 平均二乗誤差 = 0.1875 : その他

上述した条件(B1)を満たさずにロスレス16点アダマール変換を構成すると、一例として上記のような誤差が重畳するわけである。

【0048】

本実施の形態に係るロスレス16点アダマール変換をソフトウェア処理により実現するのは容易である。各々の実数型のアダマール変換部を、固定小数点演算或いは浮動小数点演算で実現し、丸め処理で切り上げを行なうところは、「0.5」を加算してからfloor関数を求め、切り捨てを行なうところは、何も加算せずにfloor関数を求めることで、図5の構成と同等の演算をソフトウェア処理で実現できる。

【0049】

<実施の形態2>

上記実施の形態1で実現したロスレス16点アダマール変換において、入力データを4×4の行列に規則的に並べると、4×4の2次元データに対するロスレス2次元アダマール変換になる。2次元的に処理する構成を示す前に、その準備として数学的な表記が下記(6)式のように簡潔になることを示す。

【0050】

【数6】

$$\left[\begin{bmatrix} + & - & - & - \\ - & + & - & - \\ - & - & + & - \\ - & - & - & + \end{bmatrix} \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \begin{bmatrix} d0 & d4 & d8 & d12 \\ d1 & d5 & d9 & d13 \\ d2 & d6 & d10 & d14 \\ d3 & d7 & d11 & d15 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \times \frac{1}{2} \begin{bmatrix} + & - & - & - \\ + & - & + & + \\ + & + & + & - \\ + & + & - & + \end{bmatrix} \right] \quad \dots \quad (6)$$

【0051】

この(6)式において、「+」と「-」の要素から成る1つ目(一番左)と5つ目(一番右)の行列は、いずれも丸め処理の内容を表わす。ここで、各記号の意味は図4における「+」と「-」と同じである。左から2つ目と3つ目の行列間の演算をした後、係数1/2を掛けて、最初の行列で定義される丸め処理を行ない、その後4つ目の行列との間で演算を行ない、係数1/2を掛けた後、最後の行列の丸め処理を行なう。非線形演算では演算の順序が変わると特性が変わるので、この場合、一番左の行列の丸め演算を先に行なわないと、平均誤差が「0」にならない。

【0052】

但し、この(6)式の変換は一例であって、平均誤差が「0」になるものは、全部で512種類あることは既に述べた。それらの違いは、丸め処理の内容を表わす一番左と一番右の行列の違いのみである。よって、その2つの行列を表記すれば、ロスレス4×4の2次元アダマール変換(ロスレス16点アダマール変換)を規定することができる。

【0053】

一方、このロスレス 16 点アダマール変換を 2 次元的に処理する構成は図 9 のようになる。ここではデータを 4 つずつ垂直方向に処理して、その結果を転置後、水平方向に処理するものである。処理の順序は逆でもよい。

【0054】

図 9 は、本発明の実施の形態 2 に係るロスレス 2 次元アダマール変換の構成を示す図である。

【0055】

図 9 において、901, 911 は、前述の (3) 式又は (4) 式の変換行列に基づいてアダマール変換処理をするアダマール変換部である。903, 913 はデコーダで、処理する 2 次元データの列番号または行番号に基づき、切り上げ処理用の加算データを生成する。905, 915 は加算部で、切り上げ用の加算処理を行なう。907, 917 は小数部を切り捨てる切り捨て部、909 はバッファで、 4×4 の 2 次元データを一時的に格納し、その転置（並び替え）を行なうのに使用される。

【0056】

尚、ここで小数部を切り捨てる切り捨て部 907, 917 は、処理の概念上設けたもので、実体は小数部の信号線をそこで断線して整数部のみを出力することで実現できる。

【0057】

まず、1 列目の 4 つのデータを並列にアダマール変換部 901 へ入力する。ここでは、前述の (3) 式又は (4) 式の変換行列に従って変換処理を行なう。ここでは、変換行列によって逆変換側の丸め処理が変わってくるので、(4) 式のウォルシュアダマール変換行列を用いることとする。このアダマール変換部 901 で変換処理されたデータは、1 ビットの小数部を持つデータとなり、加算部 905 に送られる。

【0058】

一方、デコーダ 903 は、入力される列番号情報に基づいて、切り上げ処理用の加算データを生成して加算部 905 に供給する。即ち、列番号 1, 2, 3, 4 に対して、各加算データ (1, 0, 0, 0), (0, 1, 0, 0), (0, 0,

1, 0), (0, 0, 0, 1) を生成して加算部 905 へ送る。これら加算データは、前述の (6) 式における一番左の行列の列ベクトルに対応しており、「+」を「1」に、「-」を「0」に置き換えたものである。

【0059】

この切り上げ用の加算データは、加算部 905 にて変換データの小数部に加算される。よって、加算データの重みを「0.5」として加算することになる。そして次の切り捨て部 907 にて小数部を切り捨てることにより丸め処理が完結する。ここで、変換データの小数部が「0」である場合、加算部 905 と切り捨て部 907 で行なわれる丸め処理は、該変換データに何の影響も与えないが、該変換データの小数部が「1」である場合には、「0.5」を加算した変換データは切り上げられ、その他のデータは切り捨て処理されることになる。

【0060】

また、処理対象の 4 つのデータの列番号が変わると、それに対応して、上述したように切り上げ処理用の加算データも変わり、列番号毎に丸め処理が変化する。こうしてロスレスアダマール変換処理されたデータはバッファ 909 に保存される。このバッファ 909 は、4×4 データの転置を行なうだけの小規模なものであるため、例えばレジスタを用いて容易に構成できる。

【0061】

4 つの列 (列番号 4) データのロスレス変換処理が終了したら、バッファ 909 で転置して行データとして出力する。この行データは、4 点アダマール変換部 911、デコーダ 913、加算部 915 及び切り捨て部 917 を用いて、ロスレスアダマール変換される。デコーダ 913 は、(6) 式における一番右の行列の行ベクトルに対応する加算データを出力する。これにより、ロスレス 2 次元アダマール変換後の各変換データの平均誤差は「0」となり、平均二乗誤差は「0.125」になる。

【0062】

<変形例>

これら 2 つのデコーダ 903, 913 は、対応するアダマール変換部 901, 911 が出力する 4 つのデータの最下位ビット (4 つとも同じである) の値に基

づいて出力を変える構成も考えられる。即ち、デコーダ 903, 913 は最下位ビットを受け取り、その最下位ビットが「0」の時は4つとも「0」の加算データを出力し、最下位ビットが「1」の時は「0.5」又は「-0.5」を出力する。言い方を変えと、このデコーダ 903, 913 は、最下位ビット (L) を受け取り、 $(0.5 \times L)$ 又は $(-0.5 \times L)$ を出力するとも言える。このようなデコーダ 903, 913 の出力でも前述の丸め処理と同じ処理を行なうことができる。

【0063】

<実施の形態3>

この実施の形態3は、前述の実施の形態2で説明した 4×4 のロスレス2次元アダマール変換を用いたロスレス2次元DCT（離散コサイン変換）である。このロスレスDCTを最初に実現したのは、下記の文献1である。（文献1）小松邦紀、瀬崎薫：「可逆的離散コサイン変換とその画像情報圧縮への応用」、信学技報IE97-83, p.1-6, Nov. 1997

この文献1では、アダマール変換を使っていないため、乗算の回数が大変多いことが指摘されている。上記(3)式及び(4)式の変換行列を見て分かるように、4点アダマール変換では、加減算とビットシフトのみで実現でき乗算を必要としない。ロスレス変換にすると丸め処理が増えるが、本願発明人の演算法を用いれば、切り上げ処理に対応して加算を行なうだけである。

【0064】

またロスレス4点アダマール変換を使ったロスレスDCTが、下記の文献2で提案されている。但し、ここで使っているロスレス4点アダマール変換は、前述の図2の構成である。（文献2）福間慎治、大山公一、岩橋政宏、神林紀嘉：「ロスレスアダマール変換を応用したロスレス8点高速離散コサイン変換」、信学技報IE99-65, p.1-6, Dec. 1999

図10は、この文献2のロスレスDCTの構成を示すブロック図である。

【0065】

図において、1001, 1003, 1005は、ロスレス4点アダマール変換部で、他の2入力2出力の各ブロックは、2次元ベクトルを平面上で回転する処

理を行う回転ユニットである。この回転ユニットは、文献 2 にも記載されているように、Ladder Network 化することで、ロスレス回転ユニットとなる。全てのユニットがロスレス（可逆）化されれば、全体としてもロスレス（可逆）となり、ロスレス DCT 処理が可能になる。但し、この構成は一次元の変換のみを示している。

【0066】

2次元DCTを行なう場合、図11に示すように、1次元のDCTを水平方向と垂直方向に分けて処理するのが一般的である。

【0067】

図11は、従来のロスレス2次元DCTの構成を示すブロック図である。

【0068】

この演算順序では、ロスレス2次元アダマール変換を演算することができない。何故なら、水平アダマール変換と垂直アダマール変換処理が離れているからである。そのため平均誤差をゼロ化した意味がほとんどない。図11の2次元DCTの演算順序において、水平方向の後段の処理と垂直方向のロスレスアダマール変換処理を入れ替え、図12に示す順序で処理することにより、完結した2次元アダマール変換を行うことが可能となり、ここにロスレス2次元アダマール変換を使った新たなロスレス2次元DCTが実現できる。

【0069】

図12は、本発明の実施の形態3に係るロスレス2次元DCTの構成を示す図である。

【0070】

図11と図12はソフトウェアで行なう処理を処理順に並べたもので、フローチャート形式にはなっていないが、限りなくフローチャートに近いので、各々の図に対応するフローチャートを示すことは省略する。

【0071】

尚、図10に示した構成でも図12に示す順序で処理することができるが、図13に示す構成でも図12の順序で処理することができる。

【0072】

図13は、図12のロスレス2次元DCTに有効な1次元DCTの構成を示す図である。図13において、前段の2つのロスレス4点アダマール変換処理部によるロスレスアダマール変換処理は、前述の文献2の技術をそのまま採用したもので、後段のロスレス変換処理は、文献1に記載の技術を少し変形して構成したものである。図12の順序で処理した場合の効果が大変高いものである。以下に述べる変換精度の評価では、図13の構成を用いている。

【0073】

ロスレスDCTの変換精度は、互換性という点で問題となる。即ち、ロスレスDCTを用いて圧縮した符号を、従来のロッキーDCTを用いるデコーダで復号処理した場合に、圧縮による信号の劣化以外にDCTのミスマッチによる信号劣化が発生するからである。しかも、後者の信号劣化は、圧縮率が低い高画質圧縮ほど支配的になる。

【0074】

図14は、本実施の形態に係るロスレス2次元DCTの逆変換精度を示す図である。ここでは参考のために、上記文献1、2のロスレス2次元DCTの精度も比較して示している。尚、この図14に示す精度は、ITU-T H.261規格書におけるDCTの逆変換精度の仕様に基づいて計算したものである。

【0075】

図14から明らかなように、本実施の形態に係るロスレス2次元DCTは、上記文献の方法に較べて、平均二乗誤差の全画素平均と画素間の最大値が極めて近い値となっている。これは、平均二乗誤差の画素間のばらつきが少なく、全ての画素がほぼ一様に劣化することを意味している。本実施の形態における平均二乗誤差の全画素平均は、文献1、2と比較して「2/3」～「1/2」程度に減少しているだけだが、平均二乗誤差の最大値は、これら文献に比べて「1/3」～「1/3.7」以下に減少している。更に、平均誤差についても、同様な傾向がある。即ち、本実施の形態では、文献1に較べて全画素平均では2倍以上になっているが、最大値は逆に6割程度に減少している。

【0076】

このように本実施の形態によれば、全画素の平均誤差を除く他の全て項目で精

度の向上が図れた。

【0 0 7 7】

なお本発明は、複数の機器（例えばホストコンピュータ、インターフェース機器、リーダ、プリンタなど）から構成されるシステムに適用しても、一つの機器からなる装置（例えば、複写機、ファクシミリ装置など）に適用してもよい。

【0 0 7 8】

また本発明の目的は、前述した実施形態の機能を実現するソフトウェアのプログラムコードを記録した記憶媒体（または記録媒体）を、システム或いは装置に供給し、そのシステム或いは装置のコンピュータ（またはCPUやMPU）が記憶媒体に格納されたプログラムコードを読み出し実行することによっても達成される。この場合、記憶媒体から読み出されたプログラムコード自体が前述した実施形態の機能を実現することになり、そのプログラムコードを記憶した記憶媒体は本発明を構成することになる。また、コンピュータが読み出したプログラムコードを実行することにより、前述した実施形態の機能が実現されるだけでなく、そのプログラムコードの指示に基づき、コンピュータ上で稼働しているオペレーティングシステム（OS）などが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれる。

【0 0 7 9】

さらに、記憶媒体から読み出されたプログラムコードが、コンピュータに挿入された機能拡張カードやコンピュータに接続された機能拡張ユニットに備わるメモリに書込まれた後、そのプログラムコードの指示に基づき、その機能拡張カードや機能拡張ユニットに備わるCPUなどが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれる。

【0 0 8 0】

以上説明したように本実施の形態によれば、ロスレス16点アダマール変換を、 4×2 段のロスレス4点アダマール変換を用いて構成する際に、2段目の各アダマール変換に入力する4つの中間データが、奇数個は切り上げられ、残りの奇数個は切り捨てられたものになるよう1段目の丸め処理を決め、2段目の各アダマール変換は4つの入力データが同一のアダマール変換から来たものとして捉え

、それを逆変換するように丸め処理の種類を決定する。

【0081】

これにより、ロスレス変換後の各変換係数の平均誤差を「0」にすることが可能となった。また、各変換係数の平均誤差が「0」となる 4×4 データに対するロスレス2次元アダマール変換を実現できた。

【0082】

更には、ロスレス2次元DCTの演算過程で、前記 4×4 のロスレス2次元アダマール変換を行なうことにより、該ロスレス2次元DCTの変換精度を従来方式より上げることができた。

【0083】

また、2次元処理のロスレス変換装置では、処理対象2次元データの行番号もしくは列番号に基づいて丸め処理が変更できる構成にすることにより変換精度を上げることができる。

【0084】

更には、ロスレス2次元DCTの演算過程で、ロスレス2次元アダマール変換を行なうことにより、変換精度のよいロスレス2次元DCTを実現できる。

【0085】

【発明の効果】

以上説明したように本発明によれば、ロスレスアダマール変換処理によって発生する誤差が理論上ゼロになるように丸め処理の内容を定めることにより変換係数の精度を高めることができる。

【図面の簡単な説明】

【図1】

従来の 4×4 の整数型可逆アダマール変換処理を実現する基礎となるシグナルフローを示す図である。

【図2】

従来の 4×4 の整数型可逆アダマール変換処理のシグナルフローを示す図である。

【図3】

従来のロスレス 16 点アダマール変換の構成を示すブロック図である。

【図 4】

本願発明者が既に提案した 8 種類のロスレス 4 点アダマール変換を説明する図である。

【図 5】

本発明の実施の形態 1 に係るロスレス 16 点アダマール変換の構成を示すブロック図である。

【図 6】

本実施の形態 1 に係るロスレス 16 点アダマール変換処理における変換誤差の分布を説明する図である。

【図 7】

本実施の形態に係る図 5 のロスレス 16 点アダマール変換との対比のために、変換時の平均誤差が 0 にならない構成の一例を説明する図である。

【図 8】

図 7 の構成における変換誤差の分布を説明する図である。

【図 9】

本発明の実施の形態 2 に係るロスレス 2 次元アダマール変換の構成を示すブロック図である。

【図 10】

文献 2 に記載されたロスレス DCT の構成を説明する図である。

【図 11】

従来のロスレス 2 次元 DCT の構成を示すブロック図である。

【図 12】

本発明の実施の形態 3 に係るロスレス 2 次元 DCT の構成を示すブロック図である。

【図 13】

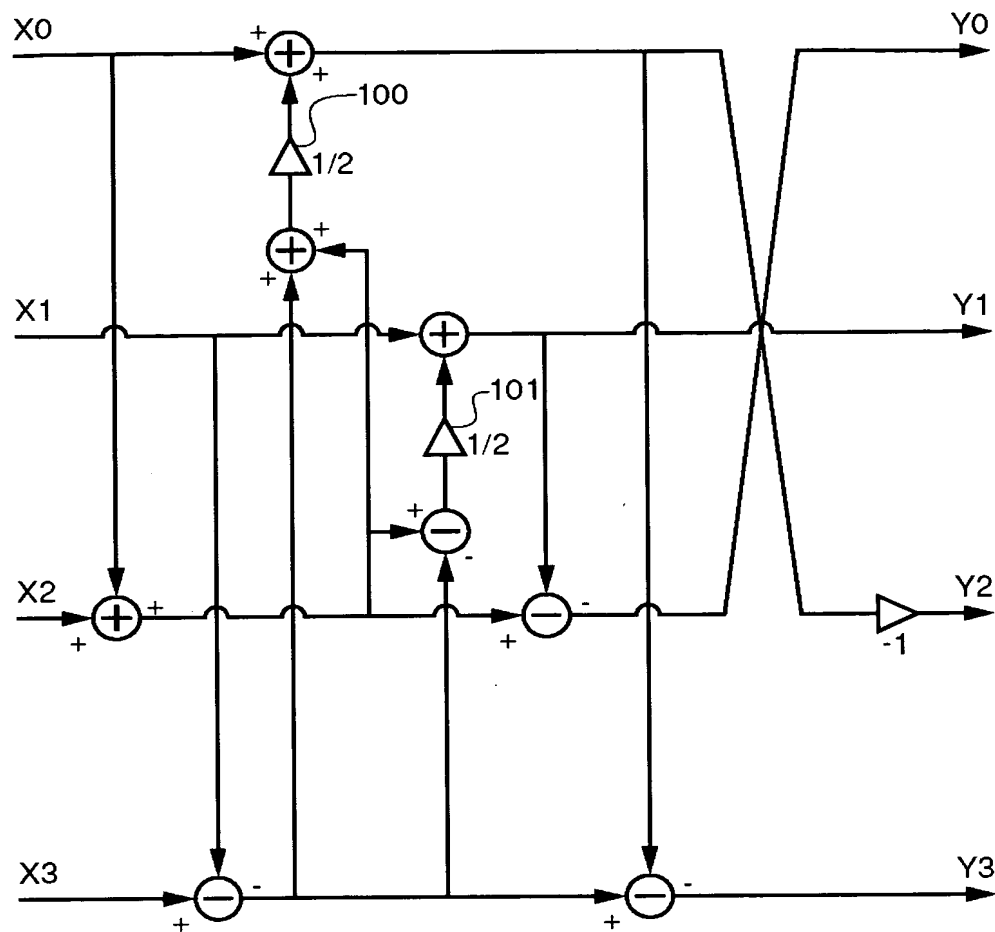
本発明の実施の形態 3 に係るロスレス 2 次元 DCT に有効な 1 次元 DCT の構成を示す図である。

【図 14】

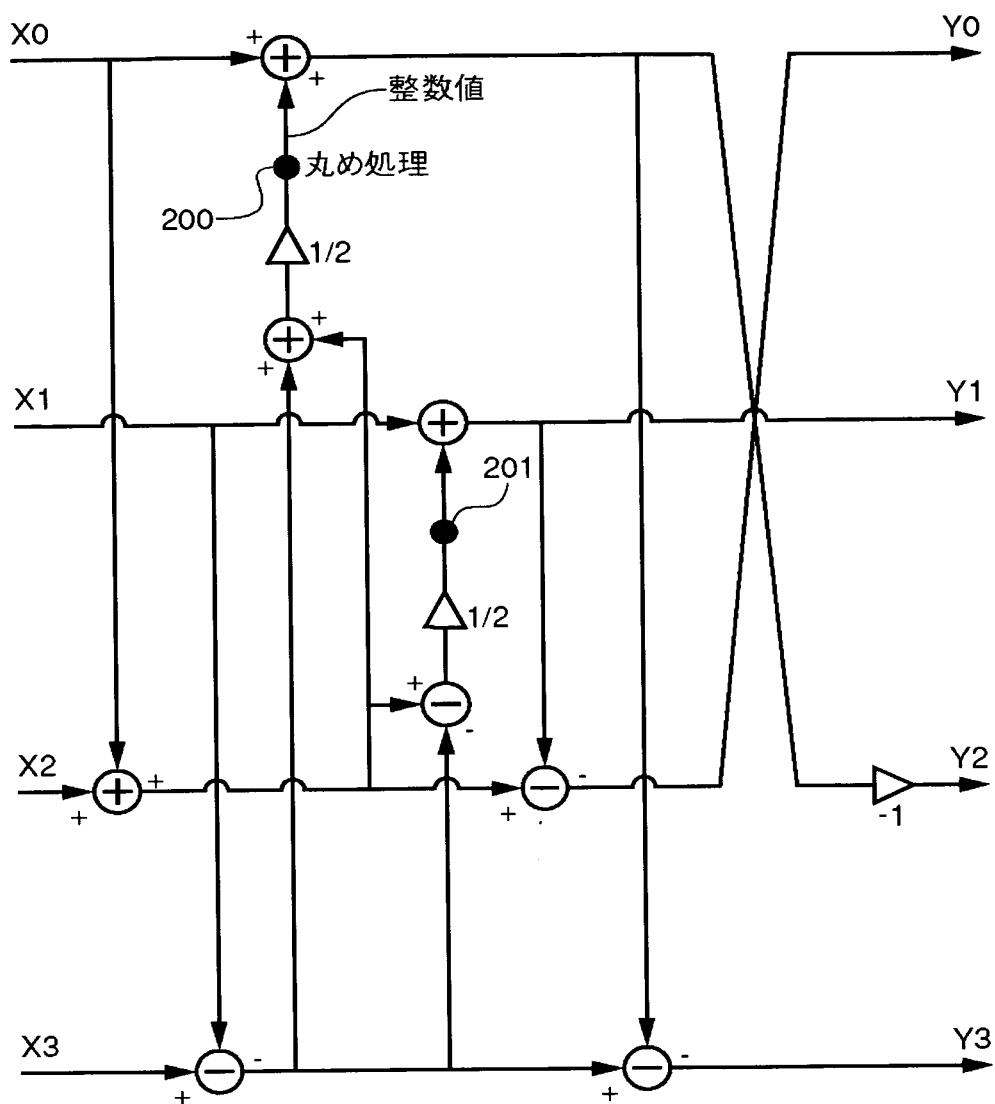
ロスレス 2 次元 D C T の逆変換精度を比較して示す図である。

【書類名】 図面

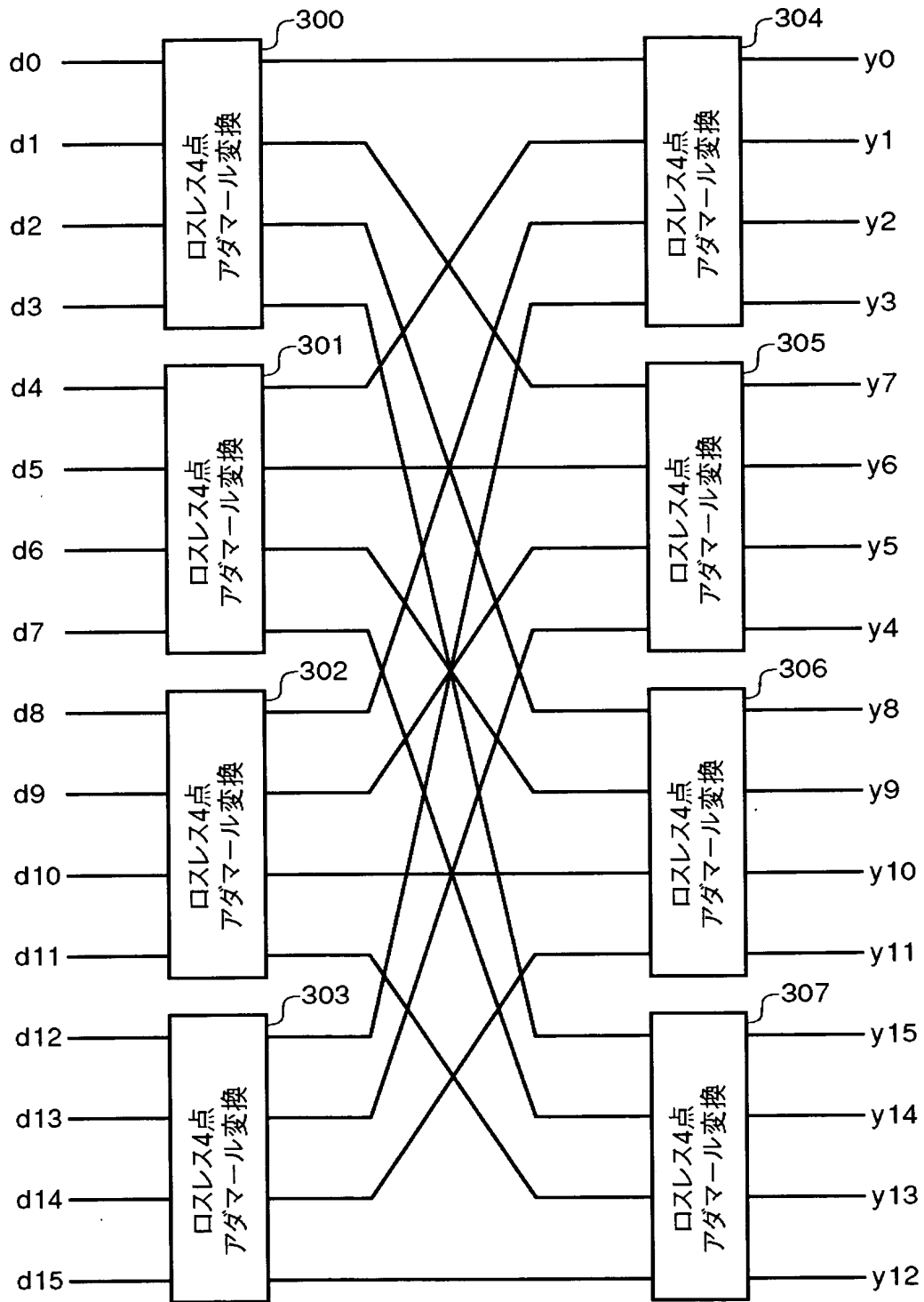
【図 1】



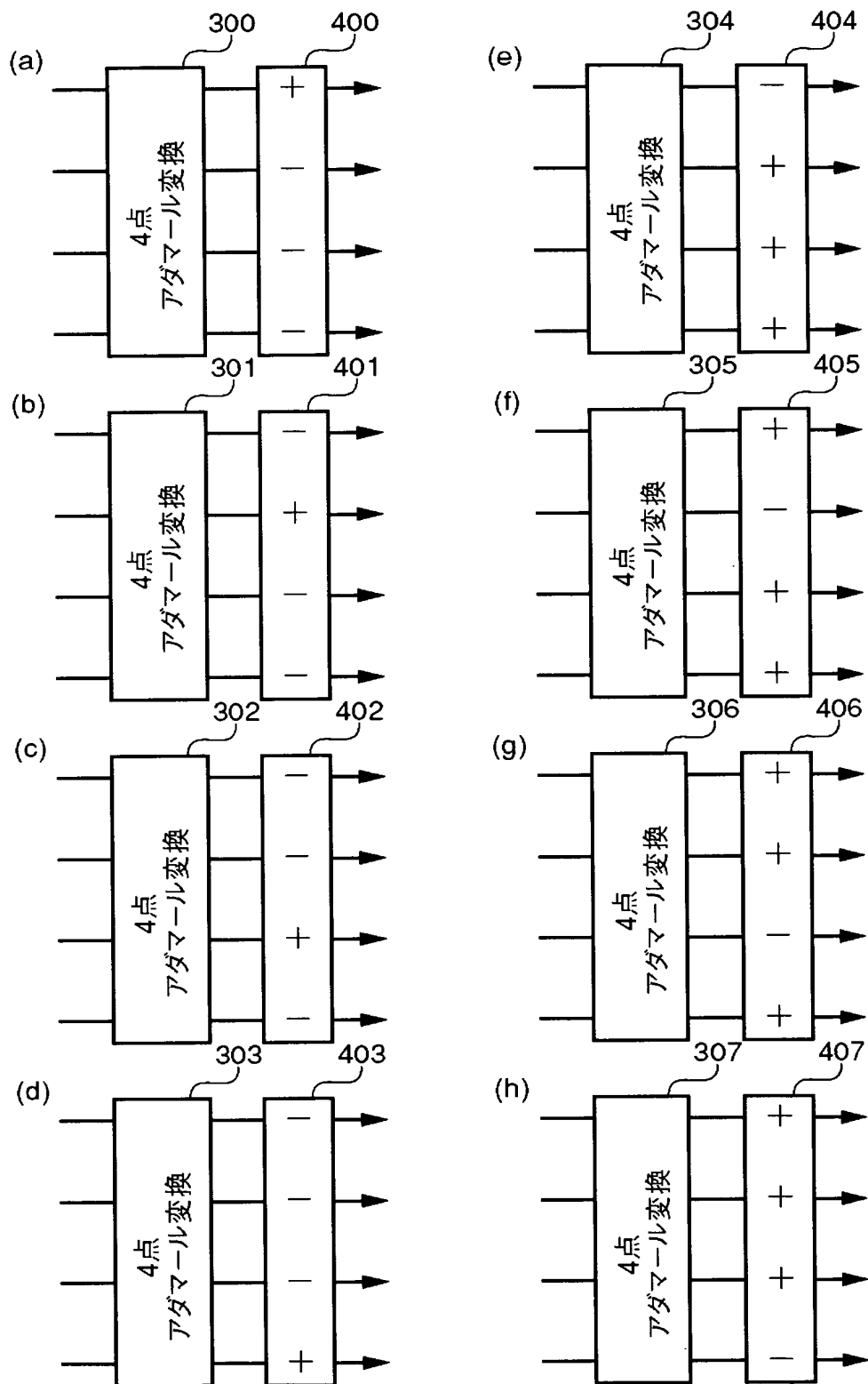
【図 2】



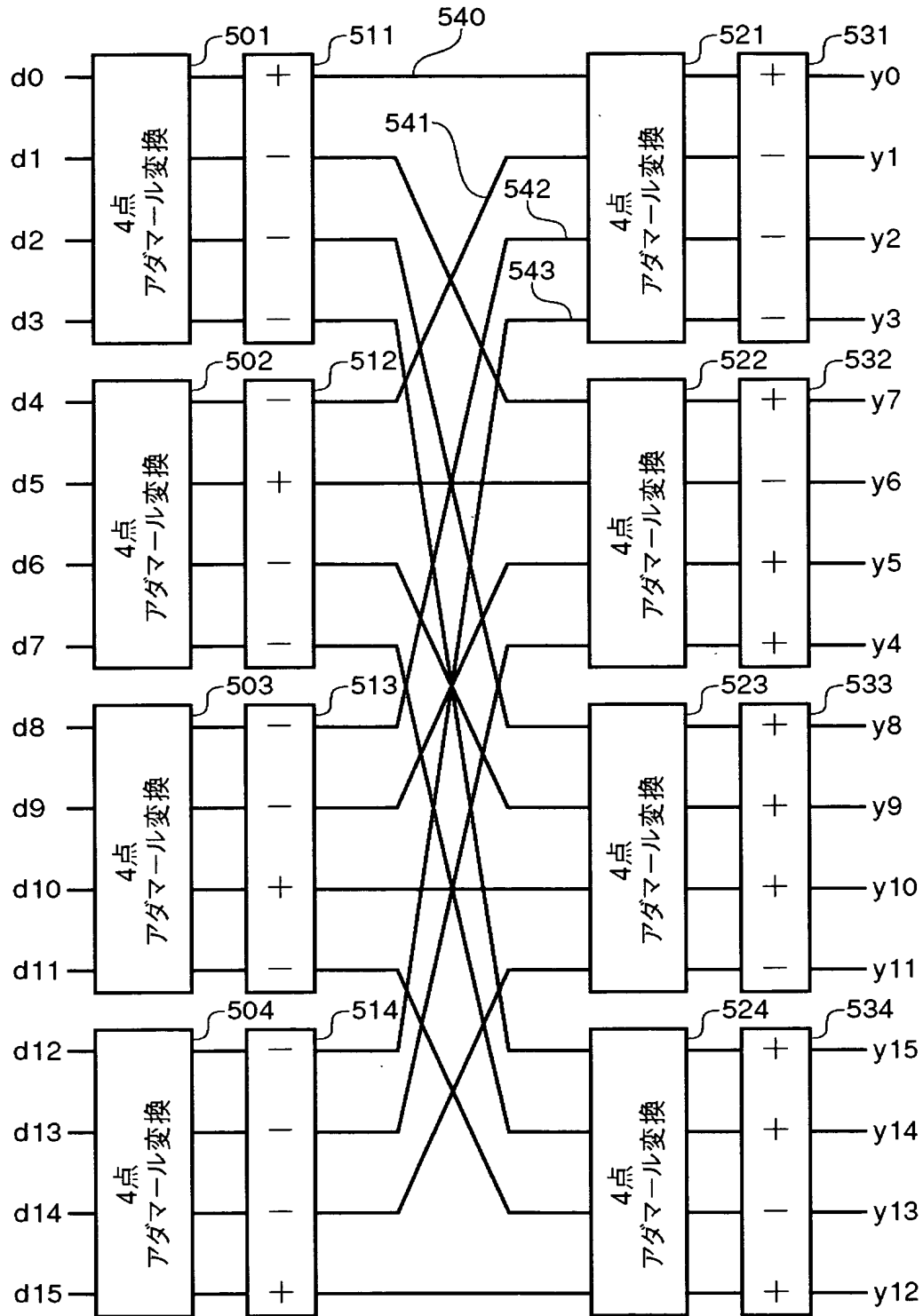
【図 3】



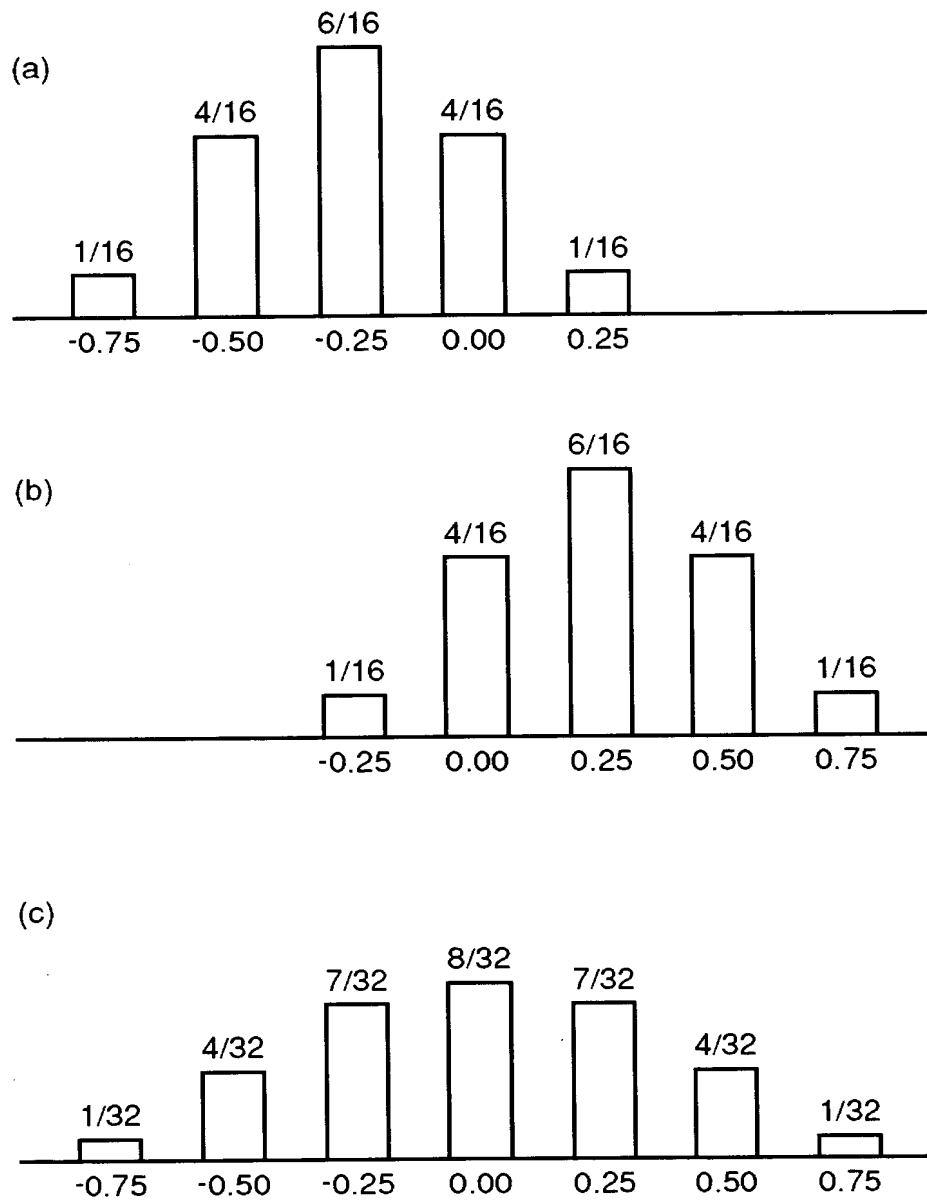
【図 4】



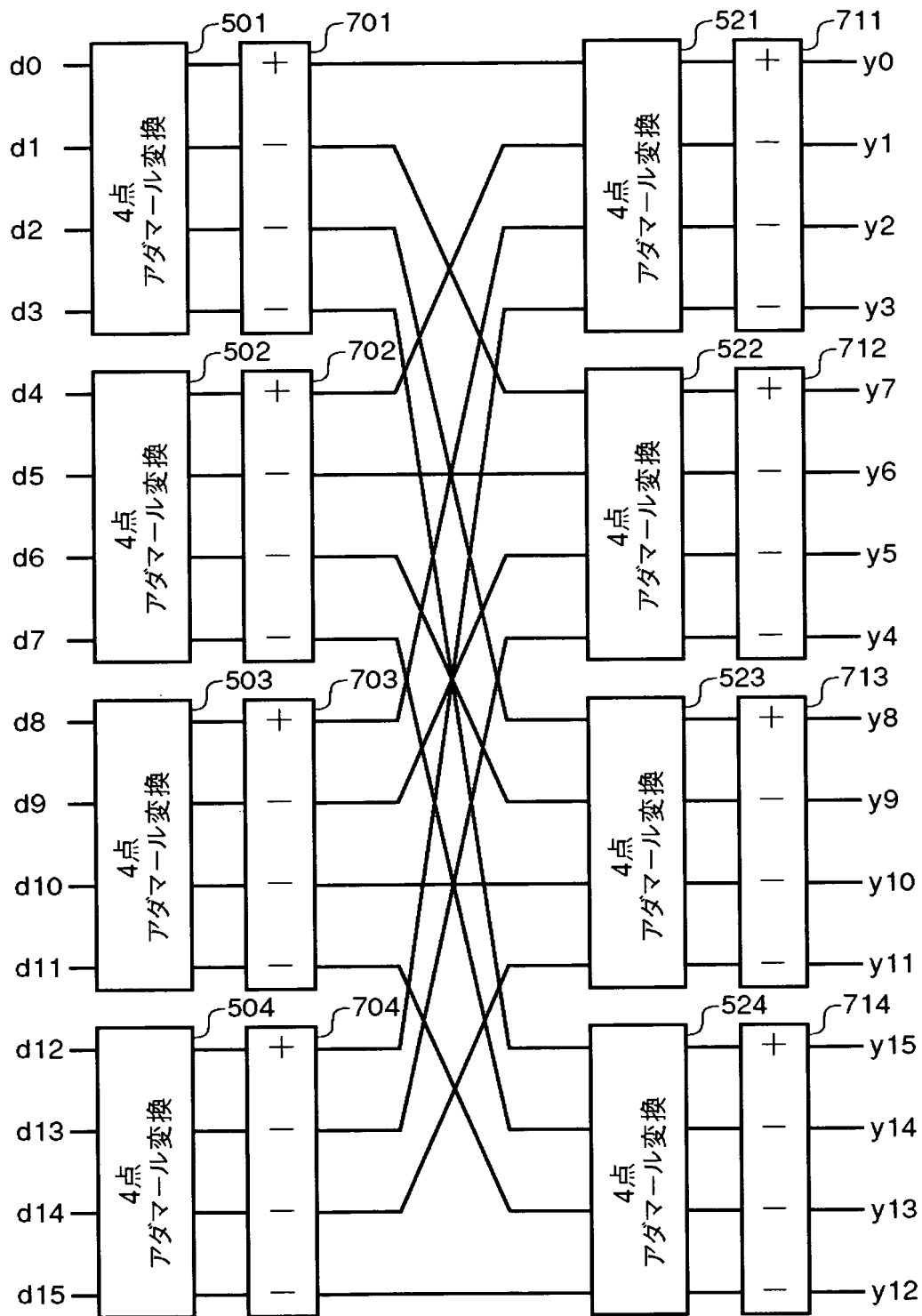
【図 5】



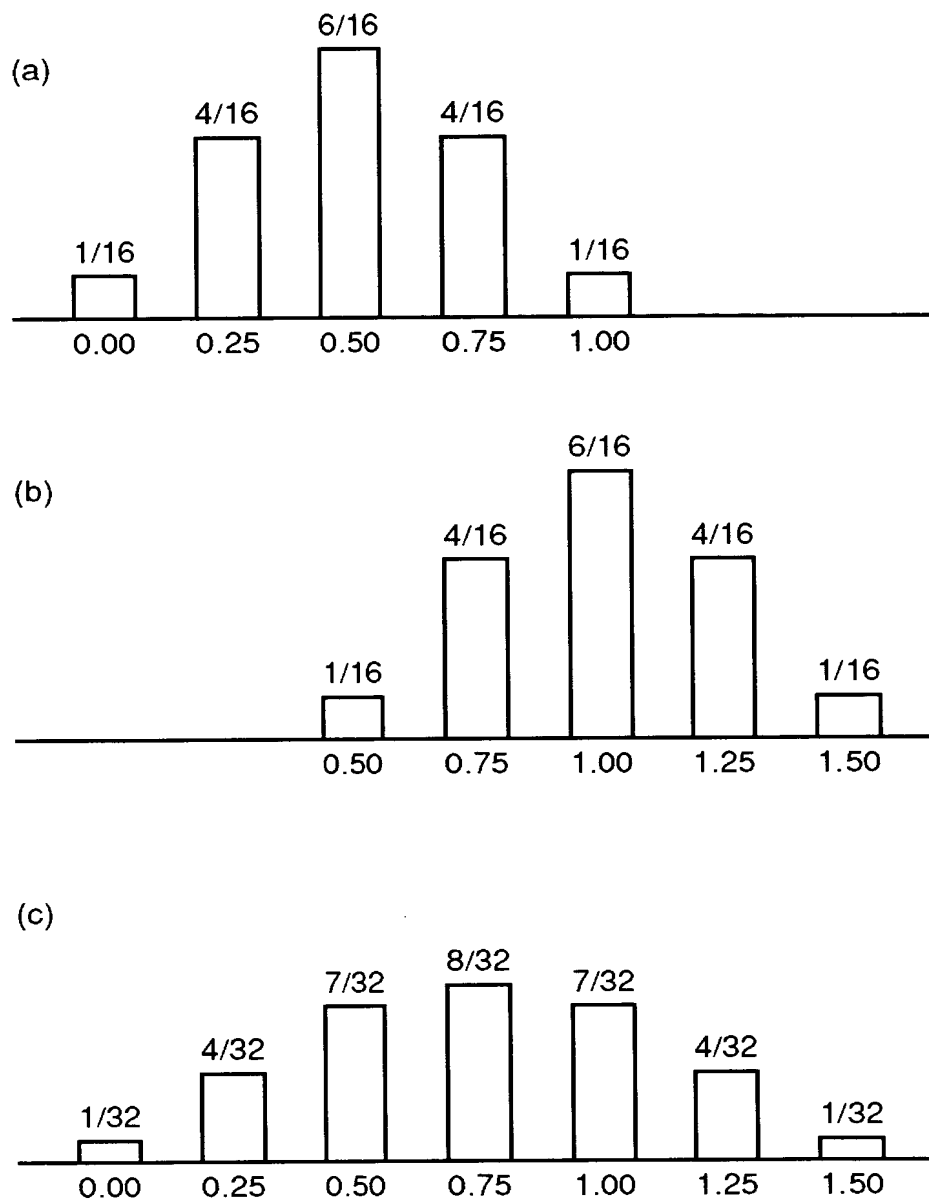
【図 6】



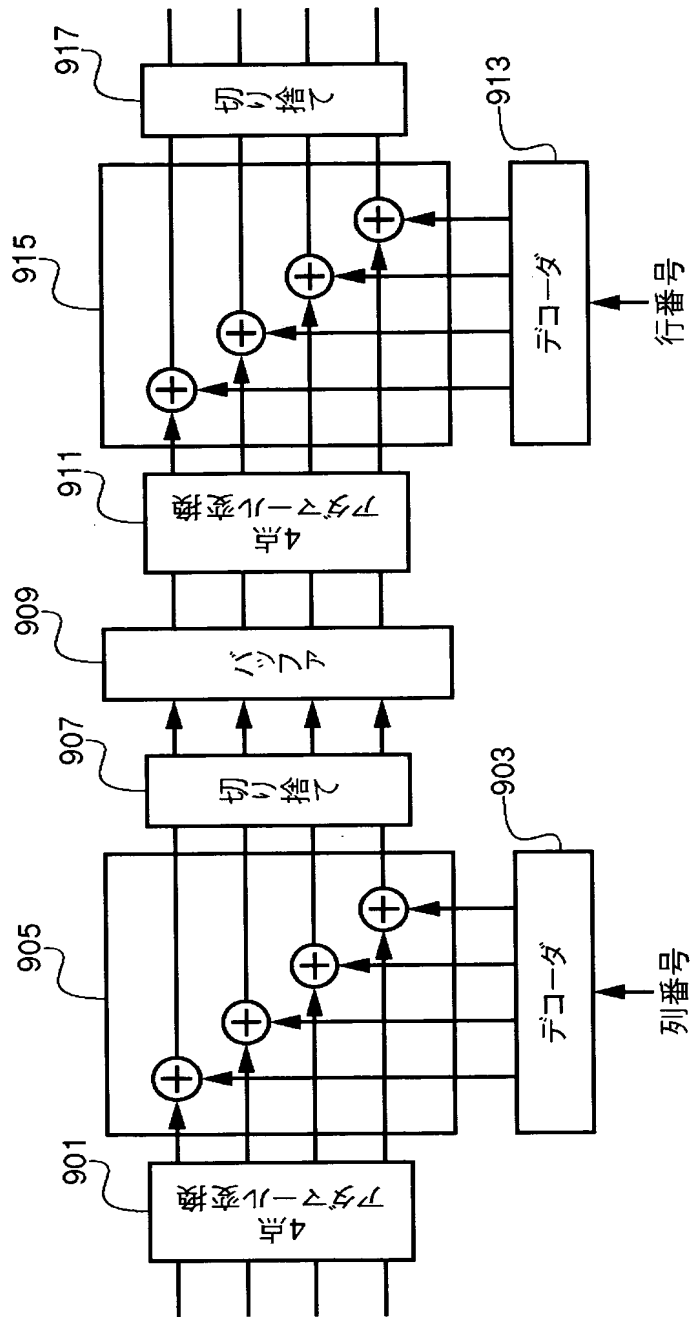
【図 7】



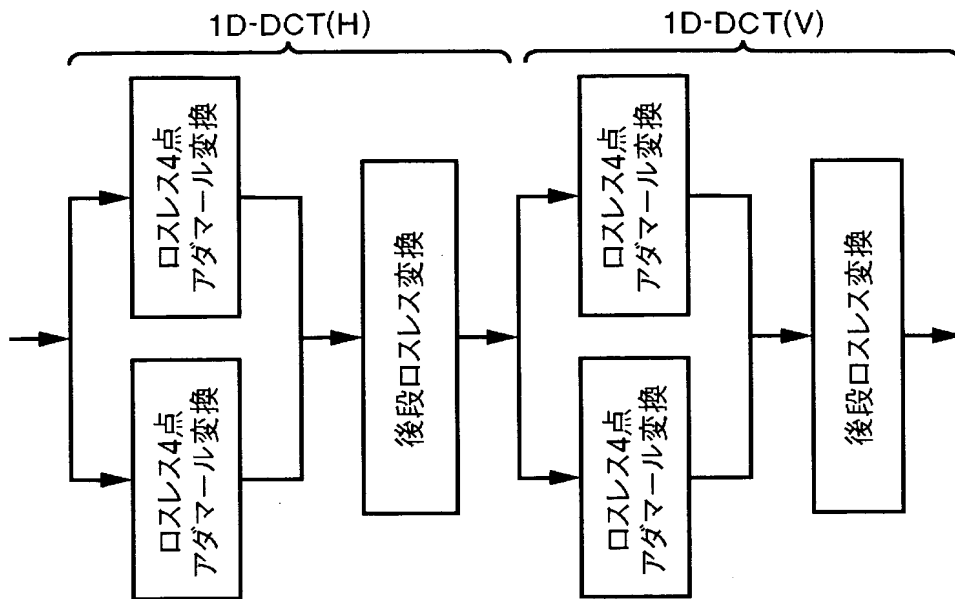
【図 8】



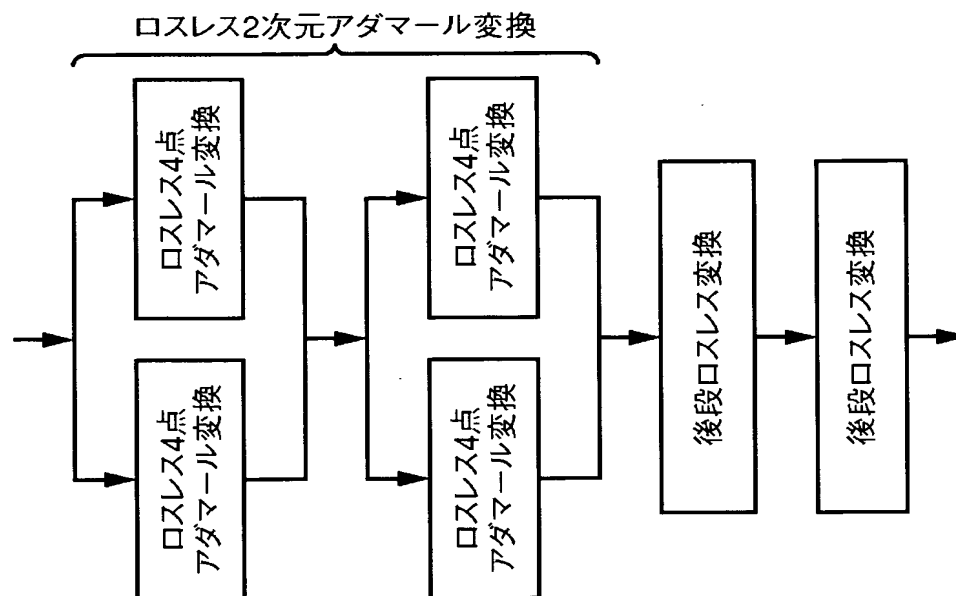
【図 9】



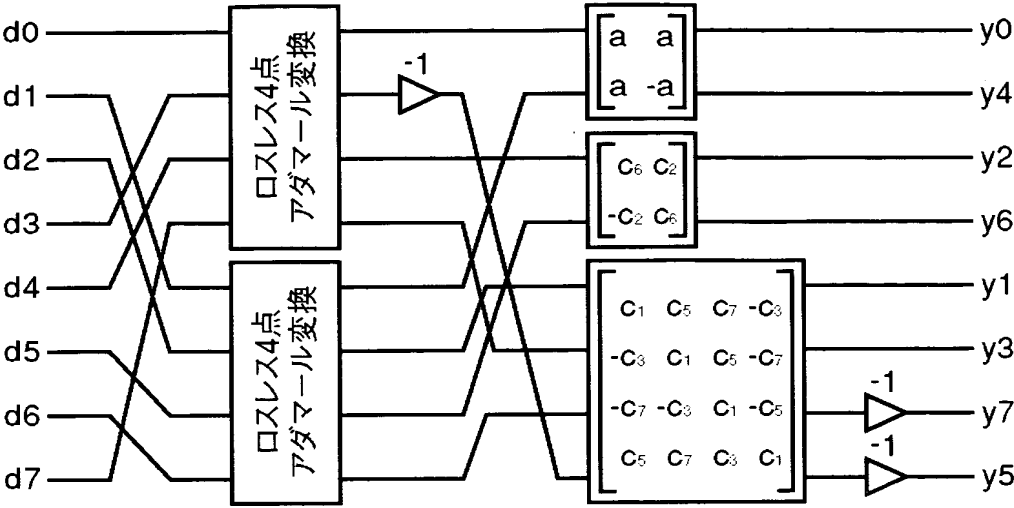
【図 11】



【図 12】



【図 1 3】



【図 1 4】

		本実施形態	文献1	文献2
最大誤差	最大値	3	-5	-4
画素毎の 平均二乗誤差	最大値	0.4693	1.5643	1.7630
	全画素平均	0.4342	1.0045	0.6478
画素毎の 平均誤差	最大値	-0.0145	0.0232	-1.0560
	全画素平均	0.0011	0.0005	0.1681

【書類名】 要約書**【要約】**

【課題】 各変換出力の誤差（数学的な変換とロスレス変換との差）の平均値や平均二乗誤差にばらつきが発生し、精度の良い変換係数と精度の悪い変換係数とができてしまう。

【解決手段】 4つの4点アダマール変換処理部501～504のそれぞれにおいて、入力信号に対してアダマール変換行列による変換処理を行い、変換処理部501～504のそれぞれから出力される4つの変換データの内、奇数個のデータの各最下位ビットを切り上げ処理し、残りの奇数個のデータの各最下位ビットを切り捨て処理して整数化する丸め処理部511～514と、4つの4点アダマール変換処理部521～524を用いて、入力信号のうち奇数個がそれと同数のロスレス4点アダマール変換処理部の出力を切り上げ処理したもの、残りの奇数個は他の残りのロスレス4点アダマール変換処理部の出力を切り捨て処理した信号を各4点アダマール変換処理部521～524に入力して変換処理を行い、各4点アダマール変換処理部521～524から出力される変換データの内、奇数個のデータの各最下位ビットを切り上げ処理し、残りの奇数個のデータの各最下位ビットを切り捨て処理して整数化する丸め処理部531～534とを有する。

【選択図】 図5

特願 2 0 0 2 - 1 9 3 2 9 8

出 願 人 履 歴 情 報

識別番号

[0 0 0 0 0 1 0 0 7]

1. 変更年月日

1 9 9 0 年 8 月 3 0 日

[変更理由]

新規登録

住 所

東京都大田区下丸子 3 丁目 3 0 番 2 号

氏 名

キャノン株式会社